

ATLAS
MANUAL DE USUARIO DEL ARQUETIPO WEB PARA PROYECTOS NEXUS

Versión 1.1

Área de Aplicaciones Especiales y Arquitectura de Software



Agencia de Informática y Comunicaciones de la Comunidad de Madrid



Hoja de Control

Título	Manual de usuario del Arquetipo Web para proyectos Nexus		
Documento de Referencia	NORMATIVA ATLAS		
Responsable	Área de Aplicaciones Especiales y Arquitectura de Software		
Versión	1.1	**Fecha Versión**	25/10/2013
[1]**Revisado/Validado por:**		**Fecha Revisión/Validación**	
[2]**Aprobado por:**		**Fecha Aprobación**	

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del documento	Unidad de Arquitectura	09/05/2012
1.1	Actualizada la url para descarga de certificados digitales de prueba	Unidad de Arquitectura	25/10/2012

Índice

1. INTRODUCCIÓN 4
1.1. AUDIENCIA OBJETIVO 4
1.2. CONOCIMIENTOS PREVIOS 5
2. REQUISITOS PREVIOS 5
3. INFORMACIÓN SOBRE EL ARQUETIPO 6
3.1. CREACIÓN DE UNA APLICACIÓN PARTIENDO DEL ARQUETIPO 6
3.2. ESTRUCTURA DEL ARQUETIPO 9
3.3. ESTRUCTURA DE DIRECTORIOS Y ARCHIVOS 9
3.3.1. Módulo web 9
3.3.2. Módulo ear 9
3.4. DESPLIEGUE Y EJECUCIÓN DE LA APLICACIÓN EN LOCAL 10
3.5. VISUALIZACIÓN DE LA APLICACIÓN DESPLEGADA EN LOCAL 10
4. DESPLIEGUE DE LA APLICACIÓN EN EL SERVIDOR DE APLICACIONES DE SAP 12
4.1. GENERACIÓN DE LOS PROYECTOS PARA EL EAR Y LA APLICACIÓN WEB 12
4.2. APERTURA DE LOS PROYECTOS DESDE NETWEAVER DEVELOPER STUDIO 13
4.3. DESPLIEGUE DEL EAR EN EL SERVIDOR DE APLICACIONES JAVA DE SAP 17
5. MODIFICACIONES SOBRE EL PROYECTO DE PARTIDA 21
5.1. ACTIVACIÓN DE LA INTEGRACIÓN CON ASF 21
5.2. ACTIVACIÓN DE LA INTEGRACIÓN CON BASE DE DATOS (HIBERNATE) 21
6. PREGUNTAS MÁS FRECUENTES 22
7. ENLACES RELACIONADOS 23

1. INTRODUCCIÓN

Los arquetipos son las plantillas para la generación de los distintos proyectos dentro del Framework Atlas. Estos arquetipos utilizan el plugin archetype de maven para generar la estructura de ficheros y directorios necesarios para nuestro proyecto, gestionando las librerías que le indiquemos así como las dependencias. Todas las librerías serán incluidas durante el empaquetado del proyecto, por lo que para generar y compilar un arquetipo debe estar conectado al repositorio de artefactos de la Comunidad de Madrid (Artifactory). El framework Atlas consta de los siguientes arquetipos:

- Arquetipo para inicio de proyecto
- Arquetipo para módulos de tipo web
- Arquetipo para módulos de tipo jar.
- Arquetipo para módulos de tipo webservice o servicios web
- Arquetipo para módulos de tipo batch
- Arquetipo para módulos de gestión documental (documentum)
- Arquetipo para módulos de tipo **web para proyectos Nexus**

Básicamente, el arquetipo para módulos de tipo web para proyectos Nexus es un arquetipo web al que se han realizado las siguientes modificaciones:

1) **Eliminación del submódulo "test":** que contenía las pruebas de integración utilizando Seleinum.
2) **Creación de un sumódulo denominado "ear"**: contiene todas las librerías necesarias para el proyecto, que se desplegarán utilizando el "Heavy Class Loader" del AS SAP para que todo pueda funcionar.
3) **Desactivación de la base de datos**: Por defecto, el arquetipo no requiere de ninguna conexión con base de datos, aunque se permite mediante configuración activarla si hiciese falta.
4) **Inclusión de archivos de configuración de ASF**: Para permitir a un proyecto de este tipo validar los certificados contra la plataforma ASF ejecutándose en el AS SAP.

En este documento no se repetirán todos los temas que coincidan con el arquetipo web normal, se hará referencia a la documentación de este arquetipo en los puntos donde así proceda.

Al igual que el arquetipo web, el arquetipo web para proyectos Nexus genera un **proyecto web multimodular** (esto significa que cuando lo generamos estamos generando un proyecto con varios módulos). El **submódulo web** es un proyecto de tipo war, que incluye la implementación del negocio de la aplicación y está preparado para desplegar la vista con JSF; el **submódulo ear** es un proyecto que contiene todas las librerías necesarias para el proyecto, que se desplegarán utilizando el "Heavy Class Loader" del AS SAP para que todo pueda funcionar.

1.1. AUDIENCIA OBJETIVO

Este documento está dirigido a desarrolladores de proyectos java para ICM dentro de los proyectos NEXUS, en los que se desee crear una aplicación web utilizando el framework Atlas.

1.2. CONOCIMIENTOS PREVIOS

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

- Java
- Eclipse
- Maven
- Spring Framework.
- Hibernate
- JSF
- Entorno de desarrollo **SAP Netweaver Developer Studio**

2. REQUISITOS PREVIOS

Para poder seguir este manual y ejecutar aplicaciones ATLAS en el servidor AS JAVA SAP, es necesario cumplir los siguientes requisitos:

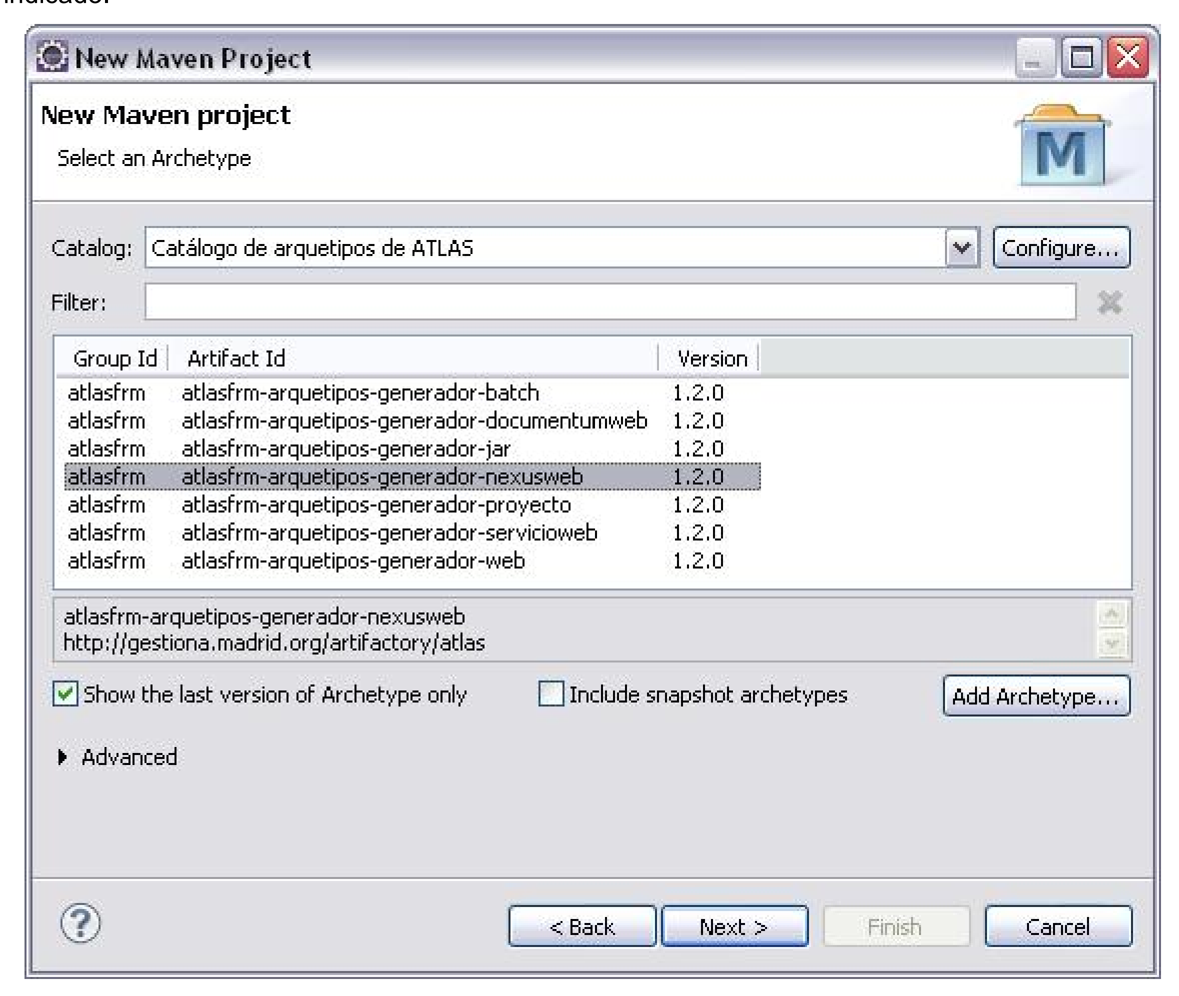
- **Versión del AS SAP**: Es **IMPRESCINDIBLE** que el servidor de aplicaciones esté instalado con al menos la versión **AS JAVA es 7.3 SP7** (se introduce una funcionalidad en este Service Pack que es necesaria).
- Tener correctamente instalado el entorno de desarrollo de ATLAS (que consiste en haber leído y seguido los pasos del **ATLAS_MUS_Preparacion_Entorno_Desarrollo**, o haber instalado el DVD con el entorno de desarrollo).
- Haber leído el manual **ATLAS_MUS_Arquetipo_Web** ya que el arquetipo de Nexus es una réplica de este con las modificaciones necesarias.

3. INFORMACIÓN SOBRE EL ARQUETIPO

3.1. Creación de una aplicación partiendo del arquetipo

NOTA IMPORTANTE
Antes de generar un arquetipo es IMPRESCINDIBLE haber generado un arquetipo para albergar el proyecto (consultar el documento "**ATLAS_MUS_Preparacion_Entorno_Desarrollo**" para ver cómo crear un arquetipo de proyecto). Cualquier arquetipo posterior será incluido como un módulo de éste.

Crear un proyecto Web para Nexus a partir de un arquetipo es similar a la creación de un proyecto Web normal, sólo hay que cambiar el nombre del arquetipo a usar. Para crear un proyecto a partir de un arquetipo Maven puede consultar la guía paso a paso que se indica en el documento **ATLAS_MUS_Preparacion_Entorno_Desarrollo**, según se indica en el apartado "Creación de una aplicación Web desde cero", **sustituyendo "atlasfrm-arquetipos-generador-web" por "atlasfrm-arquetipos-generador-nexusweb"**.
Desde Eclipse seleccionamos el arquetipo correspondiente, y seguimos con los pasos según se muestran en el manual indicado.



NOTA IMPORTANTE
Es posible que NO APAREZCA este arquetipo en el listado de arquetipos disponibles. En ese caso, hay que abrir el fichero situado en la ruta del entorno de desarrollo “C:\eclipse\archetype-catalog.xml”, y añadir las siguientes líneas dentro de la sección <archetypes>: **<archetype>** **<groupId>atlasfrm</groupId>** **<artifactId>atlasfrm-arquetipos-generador-nexusweb</artifactId>** **<version>1.2.0</version>** **<description>atlasfrm-arquetipos-generador-nexusweb</description>** **<repository>http://gestiona.madrid.org/artifactory/atlas</repository>** **</archetype>**

NOTA IMPORTANTE
Verificar que en la columna “Versión”, el número que aparece se corresponde con la versión actual de ATLAS. Si no es así, se deberá descargar la última versión del entorno de desarrollo (o modificar el fichero “C:\eclipse\archetype-catalog.xml” para introducir la última versión de ATLAS).

Antes de comenzar a visualizar la estructura del nuevo proyecto generado, debemos realizar un ajuste en el fichero ***pom.xml*** del nuevo proyecto creado. Para ello, debemos eliminar la sección ***<parent>*** existente y descomentar la sección ***<parent>*** que aparece comentada en los siguientes ficheros:

- **“\pom.xml”**
- **“*web\pom.xml*”,**
- **“ear*pom.xml*”**

Antes de eliminar:

Java EE - AtlasEjplNexus_ear/pom.xml - Eclipse

File Edit Source Navigate Search Project Run Window Help

Project Explorer | SVN Repositories | AtlasEjplNexus_ear/pom.xml

AtlasEjplNexus
ear
web
pom.xml
AtlasEjplNexus_ear
JRE System Library [JavaSE-1.6]
Maven Dependencies
src
target
pom.xml
AtlasEjplNexus_web
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.6]
Maven Dependencies
src
target
war
pom.xml

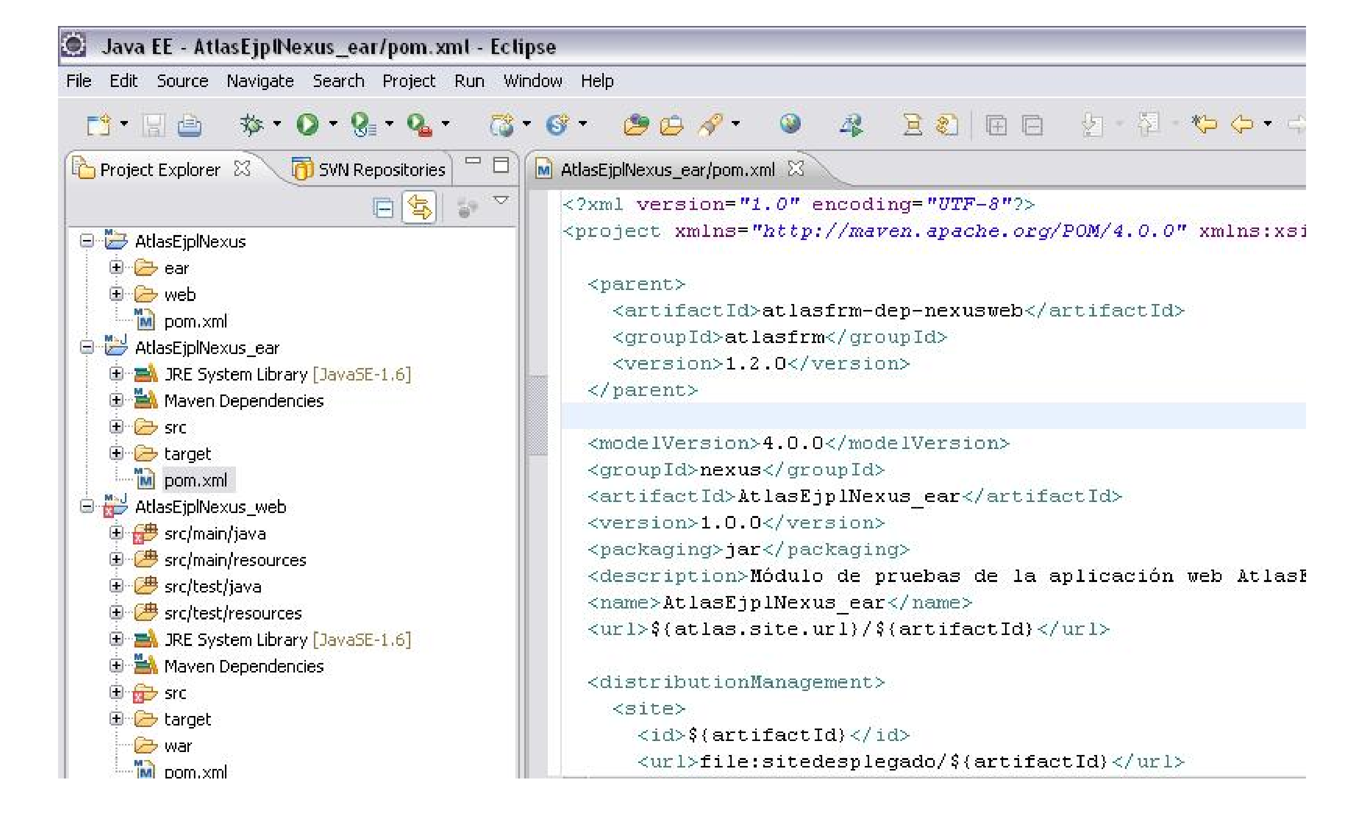
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/

<!-- Descomentar estas líneas y eliminar la :
  <parent>
    <artifactId>atlasfrm-dep-nexusweb</artif:
    <groupId>atlasfrm</groupId>
    <version>1.2.0</version>
  </parent>
-->

  <parent>
    <artifactId>AtlasEjplNexus</artifactId>
    <groupId>nexus</groupId>
    <version>1.0.0</version>
  </parent>

  <modelVersion>4.0.0</modelVersion>
  <groupId>nexus</groupId>
  <artifactId>AtlasEjplNexus_ear</artifactId:
  <version>1.0.0</version>
  <packaging>jar</packaging>
```

Después de eliminar:



NOTA IMPORTANTE
Es importante realizar esta modificación a **los tres ficheros pom.xml** indicados, si no el proyecto no funcionará.

Una realizado este cambio, ya podemos seguir viendo la estructura del subproyecto para la aplicación que acabamos de generar.

3.2. Estructura del arquetipo

La estructura del arquetipo generado es similar a la de un arquetipo Web, por lo que en este apartado sólo se muestran las diferencias entre el arquetipo Nexus Web y el Web normal.

3.3. Estructura de directorios y archivos

Una vez creado el proyecto nos generara una serie de directorios a estilo Maven, donde ubicaremos los fuentes y recursos de la aplicación web. En los siguientes apartados se muestran las diferencias principales de este arquetipo con el arquetipo Web en cada uno de sus módulos.

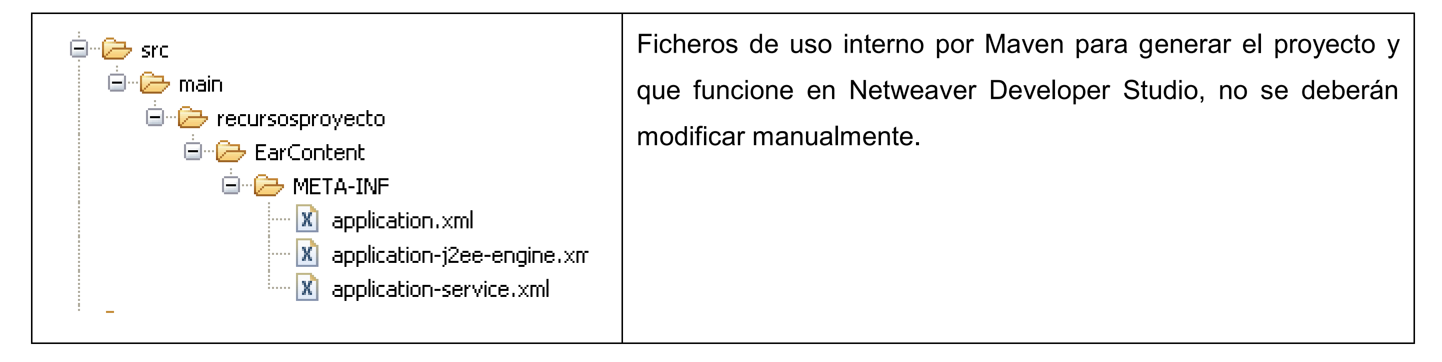
3.3.1. Módulo web

Existen las siguientes diferencias en la estructura de directorios y ficheros:

src main java recursosproyecto	En este directorio contiene ficheros que se utilizarán internamente por Maven para generar el proyecto que podrá abrirse con SAP Netweaver Developer Studio.	
src main java recursosproyecto resources	propertyFiles asf_securityagent.properties JBean.properties client-config-asf.wsdd	Ficheros utilizados para configurar la integración con la plataforma ASF para validación de certificados.
src main java recursosproyecto resources webapp	web-j2ee-engine.xml	Fichero utilizado para configuración interna del proyecto en Netweaver Developer Studio.

3.3.2. Módulo ear

Se trata de un módulo totalmente nuevo, que será el que se desplegará en el servidor de aplicaciones AS JAVA, y que contiene una referencia al submódulo web.

	Ficheros de uso interno por Maven para generar el proyecto y que funcione en Netweaver Developer Studio, no se deberán modificar manualmente.

3.4. Despliegue y Ejecución de la aplicación en LOCAL

El arquetipo web genera un proyecto multimodular, esto significa que cuando lo generamos estamos generando un proyecto con al menos dos módulos (ear y web).

Para construir el proyecto debe ejecutarse el siguiente comando **desde el directorio padre** del proyecto generado (el que contiene los dos subdirectorios “ear” y “web”):

```
mvn clean install
```

También puede ejecutarse Maven desde Eclipse. En ese caso, es importante que la casilla “Resolve Workspace artifacts” no esté activada, si lo está es posible que Maven produzca un error de ejecución del estilo: “Could not explode webapp…”.

Para ejecutar el proyecto en local podemos utilizar un servidor Jetty local que viene preconfigurado en el arquetipo. Para ejecutar directamente un servidor **jetty** con nuestro proyecto desplegado, debemos ejecutar la siguiente línea de comandos **desde el directorio web** de nuestro proyecto la siguiente línea de comandos:

```
mvn jetty:run
```

Esto arranca un servidor jetty en el puerto 9080 de nuestra máquina, en el que se encuentra desplegado el war de nuestra aplicación.

3.5. Visualización de la Aplicación Desplegada en Local

Si hemos utilizado el servidor jetty que viene preconfigurado con el arquetipo, podemos acceder a nuestra aplicación utilizando un browser, accediendo a la URL:

http://localhost:9080/secure/index.jsf

En caso de que todo haya funcionado correctamente, aparecerá un listado ejemplo:

Comunidad de Madrid - madrid.org - Microsoft Internet Explorer proporcionado por Comunidad de Madrid

http://localhost:9080/secure/index.jsf;jsessionid=1l6vbbhmsshfi

Archivo Edición Ver Favoritos Herramientas Ayuda

Comunidad de Madrid - madrid.org

Comunidad de Madrid
Aplicación de ejemplo del framework ATLAS

09 de mayo de 2012
Anónimo

Home
Menu1
Información del usuario

Inicio
Titulo
Subtitulo

Clientes

Nombre

Filtrar

Nombre	Apellidos	Dirección	Teléfono	Estado Civil	Fecha de nacimiento
nombre1	apellido10 apellido20				
nombre	apellido11 apellido21				
nombre	apellido12 apellido22				
nombre	apellido13 apellido23				
zzzzzz	apellido14 apellido24				
2nombre1	apellido10 apellido20				
2nombre	apellido11 apellido21				
2nombre	apellido12 apellido22				
2nombre	apellido13 apellido23				
2zzzzzz	apellido14 apellido24				

1 2 › » Total: 15

Nuevo Cliente

Copyright © Comunidad de Madrid
Contacto | Privacidad | Aviso Legal

Intranet local 100%

4. DESPLIEGUE DE LA APLICACIÓN EN EL SERVIDOR DE APLICACIONES DE SAP

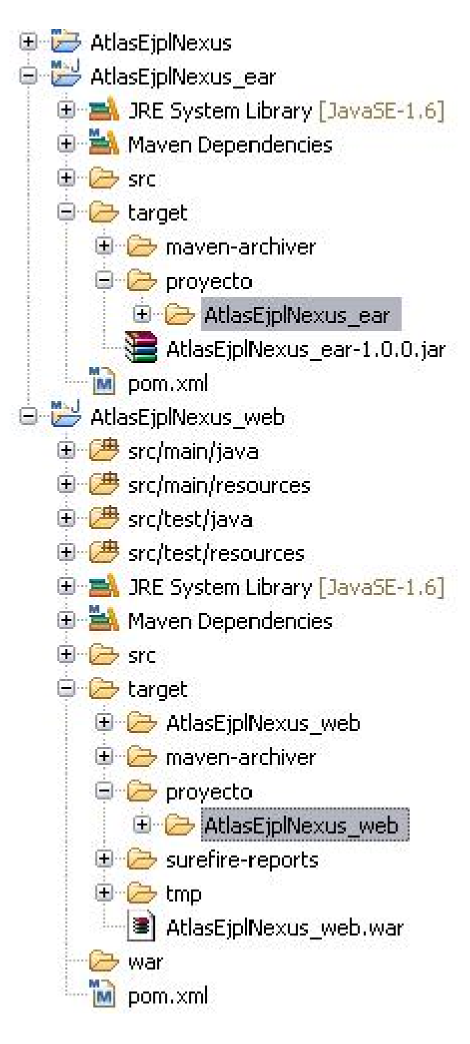
Una vez que hemos generado un proyecto partiendo del arquetipo y hemos comprobado que funciona correctamente, podemos generar un proyecto que podamos abrir desde Netweaver Developer Studio, y desde ahí desplegar la aplicación web en el servidor de aplicaciones Java de SAP. Para ello, debemos seguir los pasos indicados en los siguientes apartados.

4.1. Generación de los proyectos para el EAR y la aplicación WEB

Cuando se realizó la ejecución de "**maven clean install**" desde el proyecto padre de los dos subproyectos ear y web según se indicó en el apartado 3, automáticamente se generaron dos carpetas que contienen proyectos que pueden ser abiertos desde Netweaver Developer Studio:

- Carpeta: **ear/target/proyecto**
- Carpeta: **war/target/proyecto**

A continuación se muestra un ejemplo:



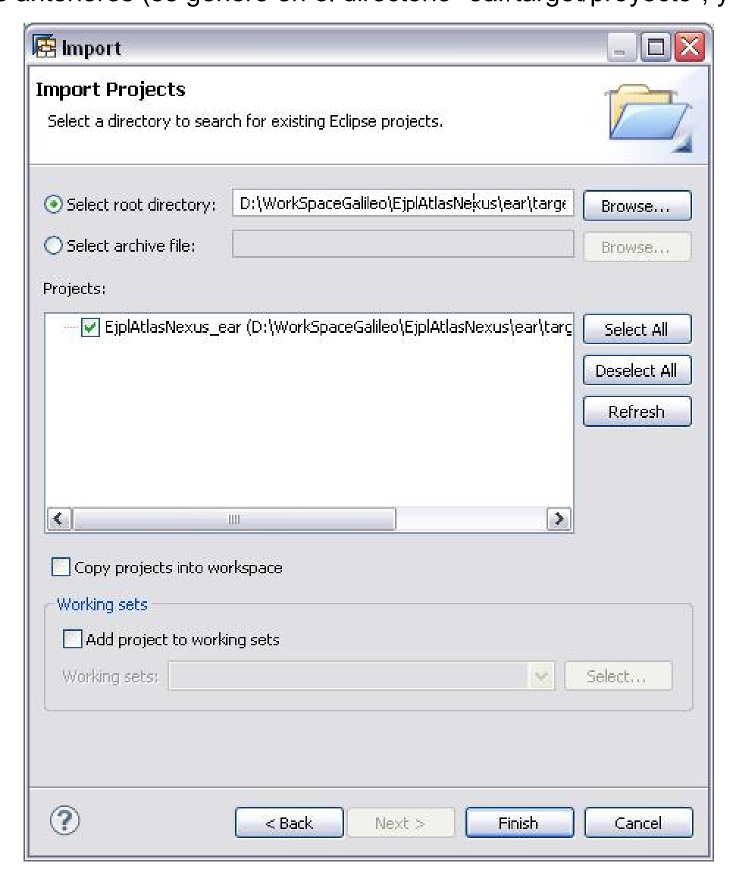
Se recomienda copiar estas carpetas a un directorio alternativo antes de proseguir con los siguientes pasos (**el directorio "target" se borra entero cada vez que se ejecuta maven**).

4.2. Apertura de los proyectos desde Netweaver Developer Studio

Para poder desplegar los proyectos en el servidor de aplicaciones Java de SAP, es necesario que abramos el entorno de desarrollo SAP Netweaver Developer Studio:

Lo primero que tenemos que hacer es abrir el proyecto generado para el **EAR**. Para abrir el proyecto debemos pulsar sobre "**File->Import->Existing Projects into Workspace**":

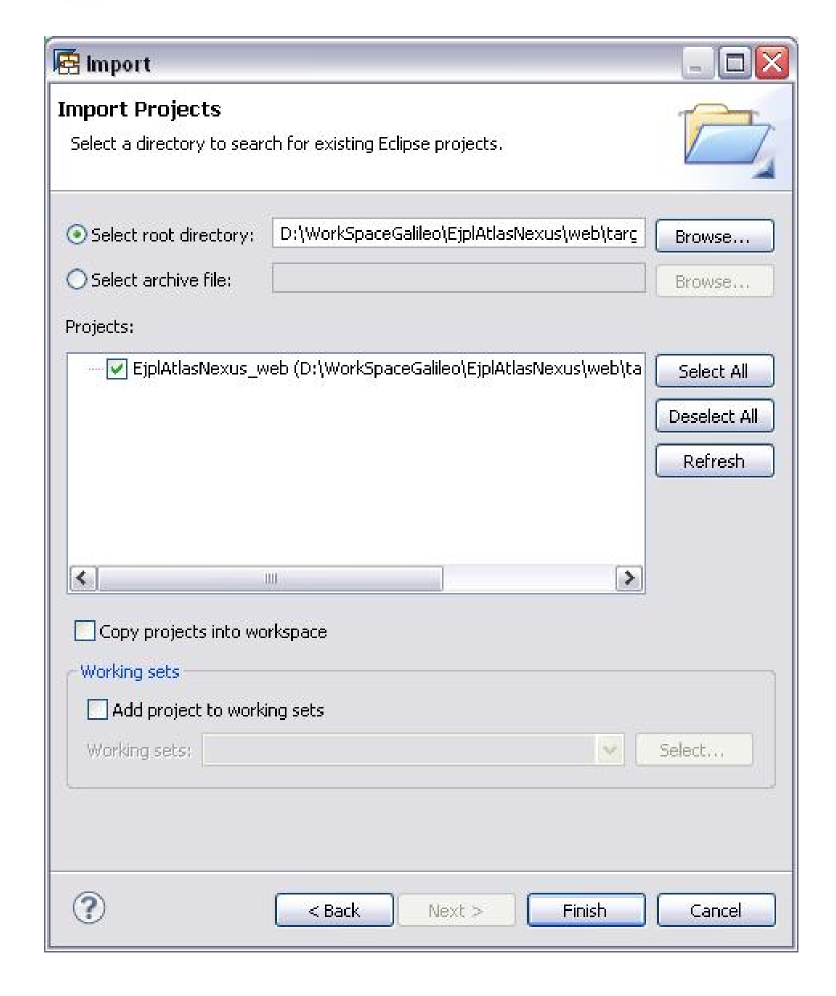
Pulsamos sobre "**Next**", y en la siguiente pantalla hacemos click en "**Browse…**" para localizar el proyecto EAR que se generó en los apartados anteriores (se generó en el directorio "ear/target/proyecto", y se hizo una copia):



Pulsamos sobre "**Finish**", y ya tenemos abierto el proyecto que representa el EAR:

NOTA IMPORTANTE
Aparecerá un error de validación de XML en el fichero "application-j2ee-engine.xml", ignorarlo pues no se trata de un error real.

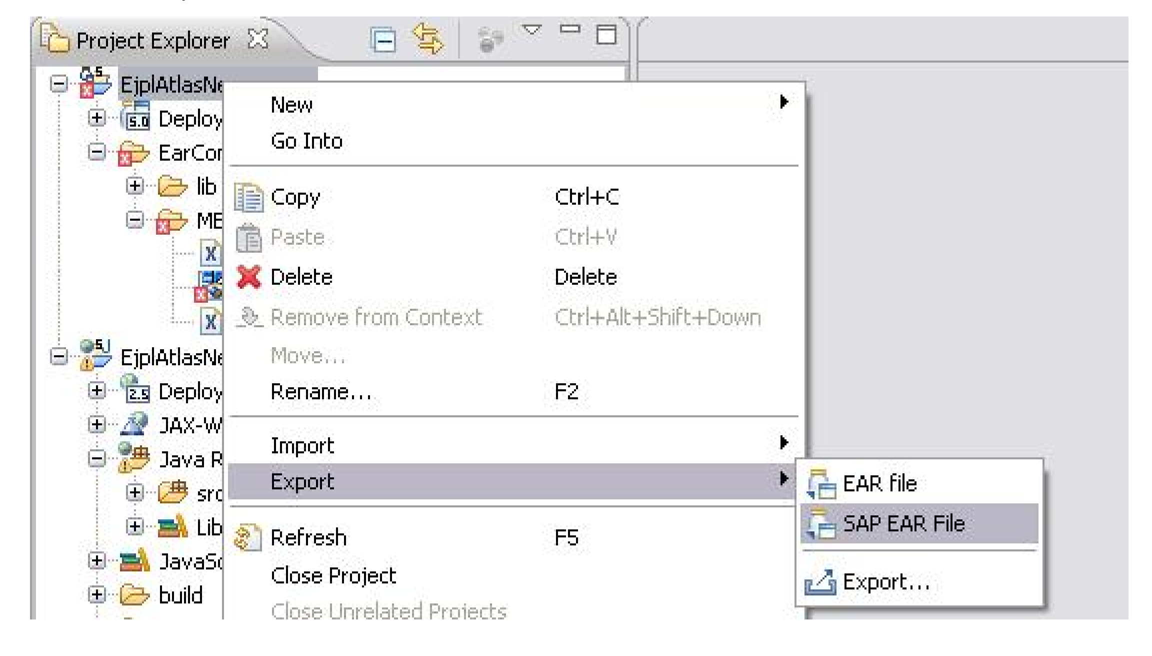
El siguiente paso es proceder a abrir el proyecto generado para la aplicación WEB. Para ello volvemos a pulsar sobre "**File->Import->Existing Projects into Workspace**", y en esta ocasión seleccionamos el proyecto correspondiente a la aplicación web que se generó en los apartados anteriores (se generó en el directorio "web/target/proyecto", y se hizo una copia):



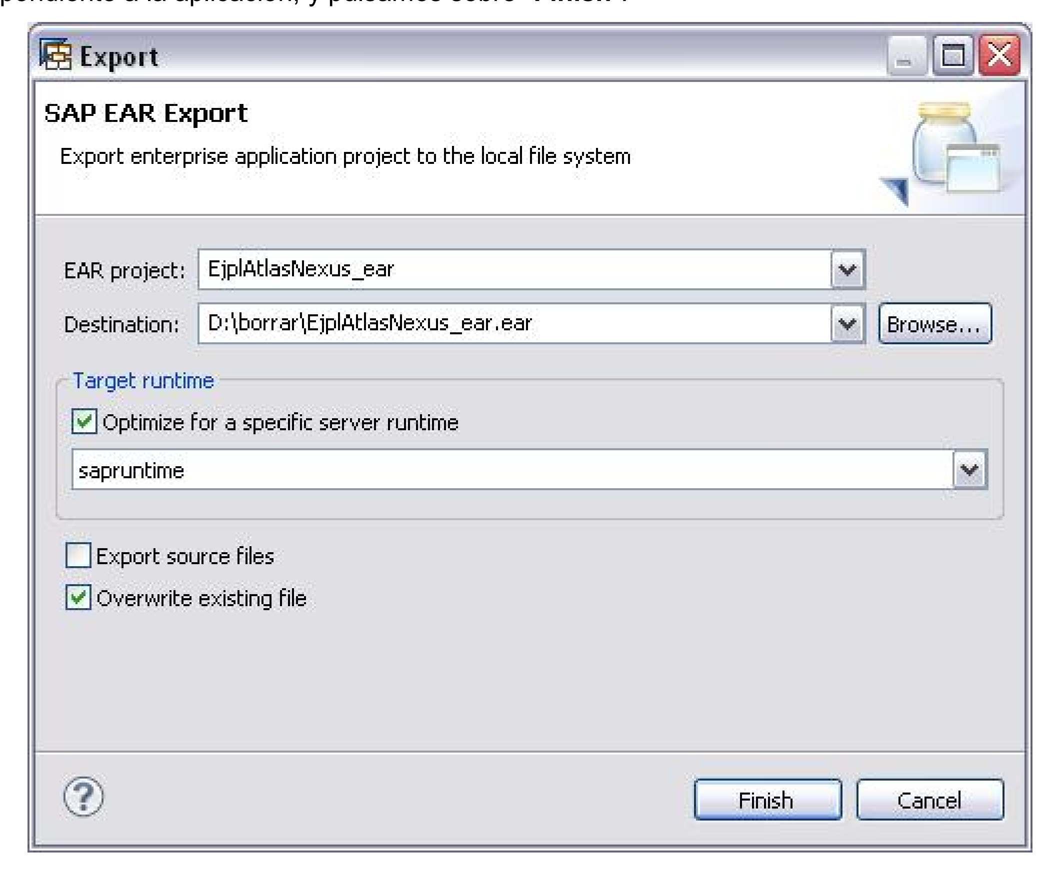
Pulsamos sobre “**Finish**”, y ya tenemos abierto ambos proyectos dentro del entorno:

4.3. Despliegue del EAR en el Servidor de Aplicaciones Java de SAP

Ya que tenemos los proyectos abiertos, antes de desplegar la aplicación tenemos que generar el fichero “EAR” asociado al proyecto. Para ello pulsamos con el botón derecho sobre el proyecto EAR y seleccionamos la opción del menú contextual “Export->SAP EAR File”:

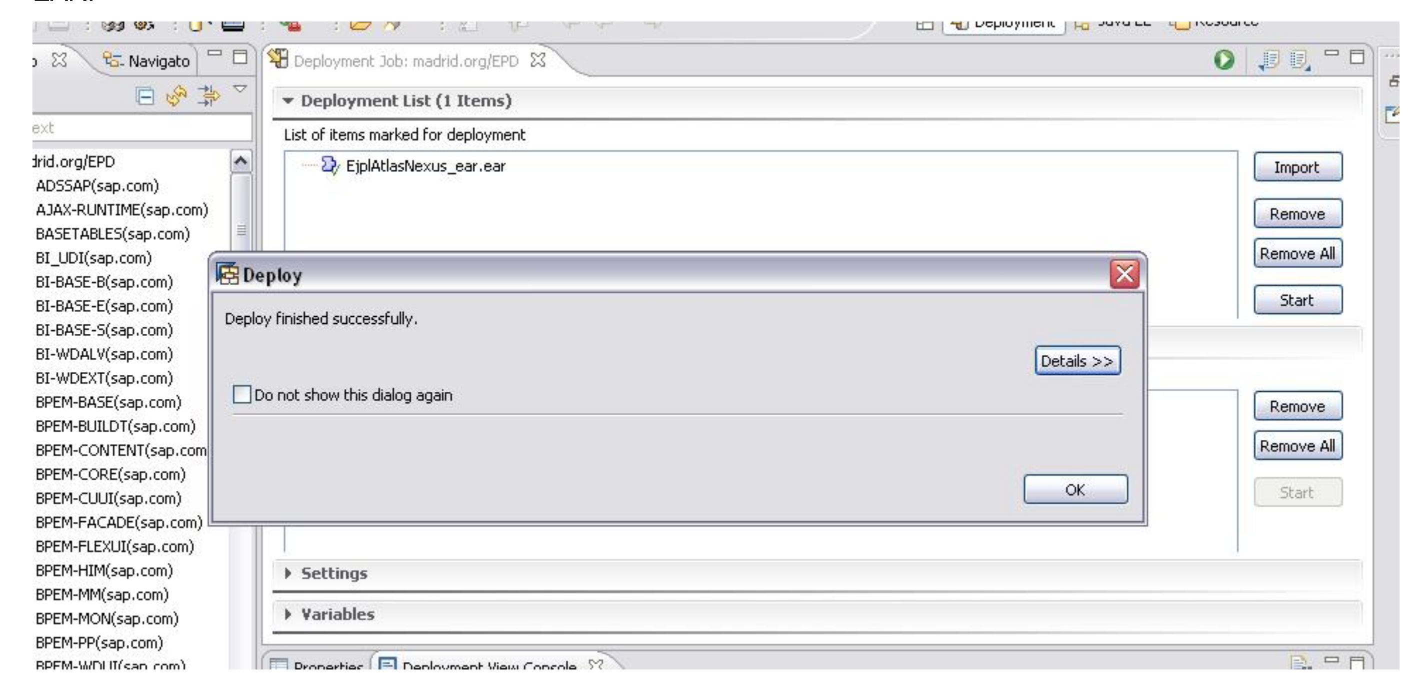


En la pantalla siguiente pulsamos sobre “**Browse**” para seleccionar el directorio donde deseamos generar el fichero “EAR” correspondiente a la aplicación, y pulsamos sobre “**Finish**”:



Una vez generado el fichero EAR, ya podemos proceder a desplegarlo. Para ello, debemos activar la perspectiva denominada “**Deployment**” del entorno:

Pulsamos sobre el botón “Import” y seleccionamos “File System” para localizar el fichero EAR que hemos generado. De vuelta a la pantalla de despliegue, pulsamos sobre el botón “Start” para realizar el despliegue del EAR:





Ya podemos acceder con cualquier Browser a la URL de la aplicación desplegada para comprobar su correcto funcionamiento:

Comunidad de Madrid - madrid

nexedepd01.madrid.org:50000/EjplAtlasNexus/secure/index.jsf

Pagina de Inicio | Google Reader | Blog de Manuel Pereir... | Analytics | My Feeds | Gmail | HOTMAIL | Twitter | Google AdWords: pág... | Otros marcadores

Comunidad de Madrid
Aplicación de ejemplo del framework ATLAS

09 de mayo de 2012
Anónimo

Home
Menu1
Información del usuario

Inicio
Titulo
Subtitulo

Clientes

Nombre

Filtrar

Nombre	Apellidos	Dirección	Teléfono	Estado Civil	Fecha de nacimiento
nombre1	apellido10 apellido20				
nombre	apellido11 apellido21				
nombre	apellido12 apellido22				
nombre	apellido13 apellido23				
zzzzzz	apellido14 apellido24				
2nombre1	apellido10 apellido20				
2nombre	apellido11 apellido21				
2nombre	apellido12 apellido22				
2nombre	apellido13 apellido23				
2zzzzzz	apellido14 apellido24				

1 2 › »   Total: 15

Nuevo Cliente

Copyright © Comunidad de Madrid   Contacto | Privacidad | Aviso Legal

5. MODIFICACIONES SOBRE EL PROYECTO DE PARTIDA

En este apartado se muestran diversas modificaciones que pueden hacerse sobre el proyecto generado a partir del arquetipo, tales como activar la integración con ASF o con base de datos a través de Hibernate.

5.1. Activación de la integración con ASF

Para activar la validación de ceritificados digitales de usuario contra la plataforma ASF es necesario seguir las indicaciones a continuación:

1) **Requisito**: Comprobar que se tiene acceso a esta URL (ASF de desarrollo) desde el servidor de aplicaciones AS JAVA: https://desarrollo3.madrid.org/asf_firma573/
2) Copiar el fichero "**WebContent/WEB-INF/clases/client-config-asf.wsdd**" a una ruta fija en la máquina del servidor de aplicaciones AS JAVA.
3) Abrir el fichero **WebContent/WEB-INF/clases/propertyFiles/asf_securityagent.properties** y:
    a. Modificar la ruta com.tbsolutions.asf.securityagent.fileclient para apuntar a la ruta del fichero client-config-asf.wsdd que se ha puesto fija en la máquina
    b. Modificar las URLs de acceso a ASF para su entorno de ASF (ahora están apuntando a desarrollo.madrid.org, si tuvieseis otro entorno hay que configurarlo aquí.
4) Editar fichero **WebContent/WEB-INF/clases/propertyFiles/JBean.properties**, para variables donde ASF vuelca el log:
    a. jfactory.util.Log.nombreFicheroLog = /usr/sap/EPD/SCS/Portal_logs/asf.log
    b. jfactory.util.Log.nombreFicheroTraza = /usr/sap/EPD/SCS/Portal_logs/asf.log.trazas
5) En el fichero **WebContent/WEB-INF/clases/environment.properites** configurar las variables:
    a. (Necesaria, Id de la aplicación en ASF) app.id.asf=APL_DEMO
    b. (Sólo para websigner) asf.clientJSUrl=https://desarrollo3.madrid.org/asf_firma573/js/WS_Full.js
    c. (Sólo para cifrado simétrico) asf.cipherKey=1gt2wAq5gh78K4rg12345678
6) En el fichero **WebContent/WEB-INF/clases/environment.properites** modificar las variables de las políticas para usar certificados con ASF:
    a. politicaMaquinaXXXXXX=politicaPublicoCertificado
7) Configurar y activar https y acceso obligado con certificado en el servidor AS JAVA.
8) Para apuntar contra ASF de desarrollo de ICM se necesita algún certificado de prueba válido en este entorno, instalado en vuestro navegador (puede descargarse el correspondiente al NIF 2W en la URL: http://intranet.madrid.org/arquitecturasw/otras-tecnologias/seguridad/certificados - en la misma página aparece el PIN para poder instalar el certificado).
9) Desplegar la aplicación, y probarla

5.2. Activación de la integración con base de datos (Hibernate)

Para activar la integración con Hibernate solicitar información a la Unidad de Arquitectura de Aplicaciones.

6. PREGUNTAS MÁS FRECUENTES

La lista de preguntas frecuentes se encuentra en el portal de arquitectura.

7. ENLACES RELACIONADOS

Producto	URL
Apache Maven	http://maven.apache.org/
Ajax4JSF	http://www.jboss.org/jbossrichfaces/
Facelets	https://facelets.dev.java.net/
Hibernate	http://www.hibernate.org/
Hibernate Annotations	http://www.hibernate.org/hib_docs/annotations/reference/en/html_single/
JSF	http://java.sun.com/javaee/javaserverfaces/
JSFUnit	http://www.jboss.org/jsfunit/
MyFaces Core	http://myfaces.apache.org/
RichFaces	http://www.jboss.org/jbossrichfaces/
Spring	http://www.springframework.org/
Spring Security	http://www.springframework.org/